Accelerating Multigrid Prolongation via Bipartite Graph Attention Networks

Mitchell T. Scott, Socratis Petrides (CASC,LLNL), Rui Peng Li (CASC,LLNL)

Abstract: When trying to solve large, sparse linear systems $\mathcal{L}u = f$, arising from an elliptic differential operator, multigrid is the go-to method as it has nice scaling properties. To convert between these grids of differing coarsening levels, the prolongation operator is used. While there is a standard technique for constructing the prolongation operator for geometric multigrid, we exploit the bipartite graph structure of the nodes. We use graph attention networks on the nodes to learn the optimal weights between the nodes. We compare this method to standard prolongation techniques on baselines from MFEM.

INTRODUCTION

Problem: Solve for u in

$$\mathcal{L}u = f + \text{b.c.}$$

where $\mathcal{L}$ is a large, sparse matrix from a partial differential equation (PDE)

Approach: Direct versus Iterative

Direct: Too large to fit in computer memory, with $\mathcal{O}(n^3)$ timing
Iterative: Approximate the solution, progressively getting better

Question: How do we approximate the solution and improve?

MULTIGRID

- Solve a system with many points? – Accurate but expensive
- Solve a system with fewer points? – Faster but inaccurate
- **Idea: Start with fine grid, coarsen to a smaller grid, and error correct**

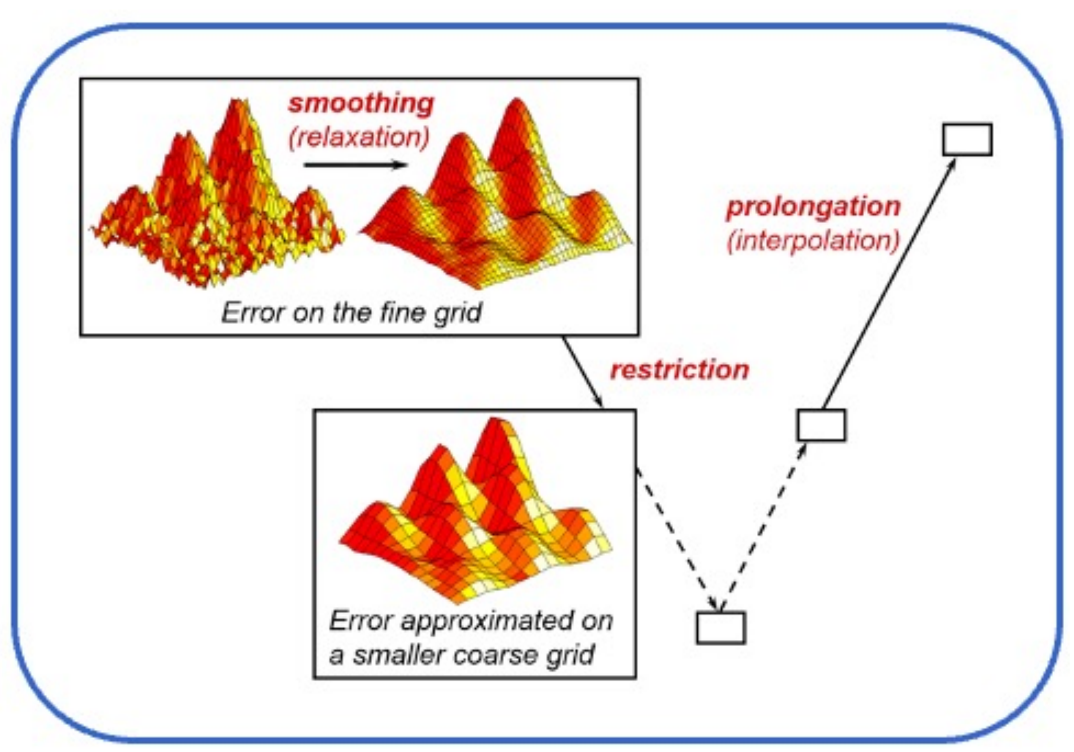

Figure 1: *V-cycle in multigrid, where the solution is found through recursively smoothing the error then restricting to a coarser grid. When the grid is small enough, a direct solve is performed. Then the solution is prolongated back to a fine grid, post-processing the error introduced through prolongation. Here there are three grid levels.*

Question: How do we perform *prolongation*, and in turn *restriction*?

MATRICES AS (bipartite) GRAPHS

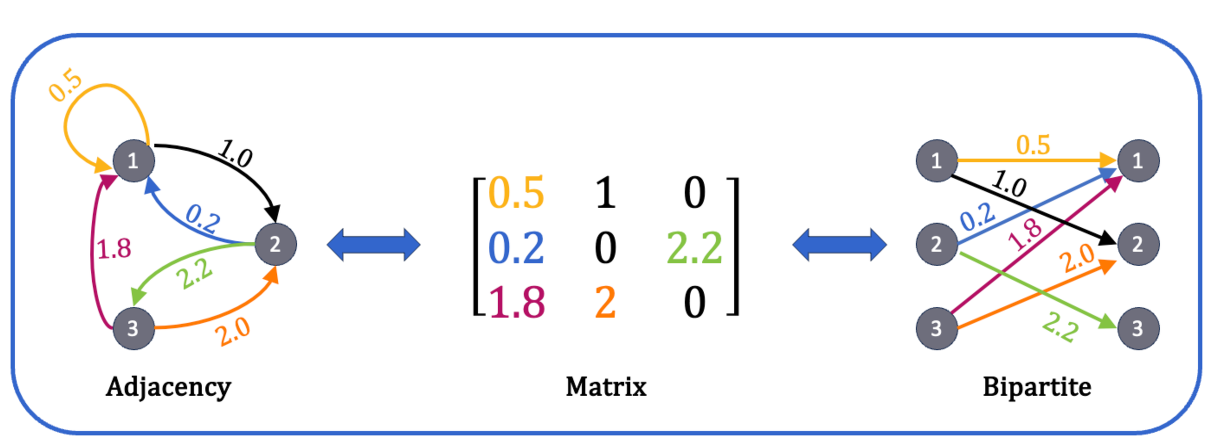

***Figure 2:* (Left)** *Adjacency graph to matrix, where rows are the start, columns are the end, and values are the weights.* **(Right)** *Bipartite graph of matrix, where the weights are left to right.*

GRAPH ATTENTION NETWORKS

- The matrix's underlying graph structure explains local information. Long-range interactions are studied by *aggregating* many *local messages*.
- Using Graph Attention Networks (GAT) [1], different weights are learned using a SoftMax for different nodes in the neighborhood:

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\boldsymbol{a}^\top[\,\mathbf{W}h_i \| \mathbf{W}h_j]))}{\sum_{k \in \mathcal{N}_i \cup \{i\}} \exp(\text{LeakyReLU}(\boldsymbol{a}^\top[\,\mathbf{W}h_i \| \mathbf{W}h_k]))}$$

- The updates efficiently put higher weights on more important neighbors through rounds of message passing.

DATA-DRIVEN PROLONGATION

- *Sparsity* is an important property to maintain for memory and speed.
- The ground truth prolongation matrix is given by MFEM.
- **Loss function:**

$$\mathcal{L}_P = \begin{cases} \left\| I - M^{-1}_{\text{Two Grid}} A \right\|, & \text{Unsupervised} \\ \| P - P_{\text{True}} \|, & \text{Supervised} \end{cases}$$

where $I - M^{-1}A = (I - S^{-1}A)(I - PA_c^{-1}P^\top A)(I - S^{-1}A)$ and $A_c = P^\top AP$

- **Model:** Graph Attention Networks – uses attention to find most important neighbors in the graph using rounds of message passing

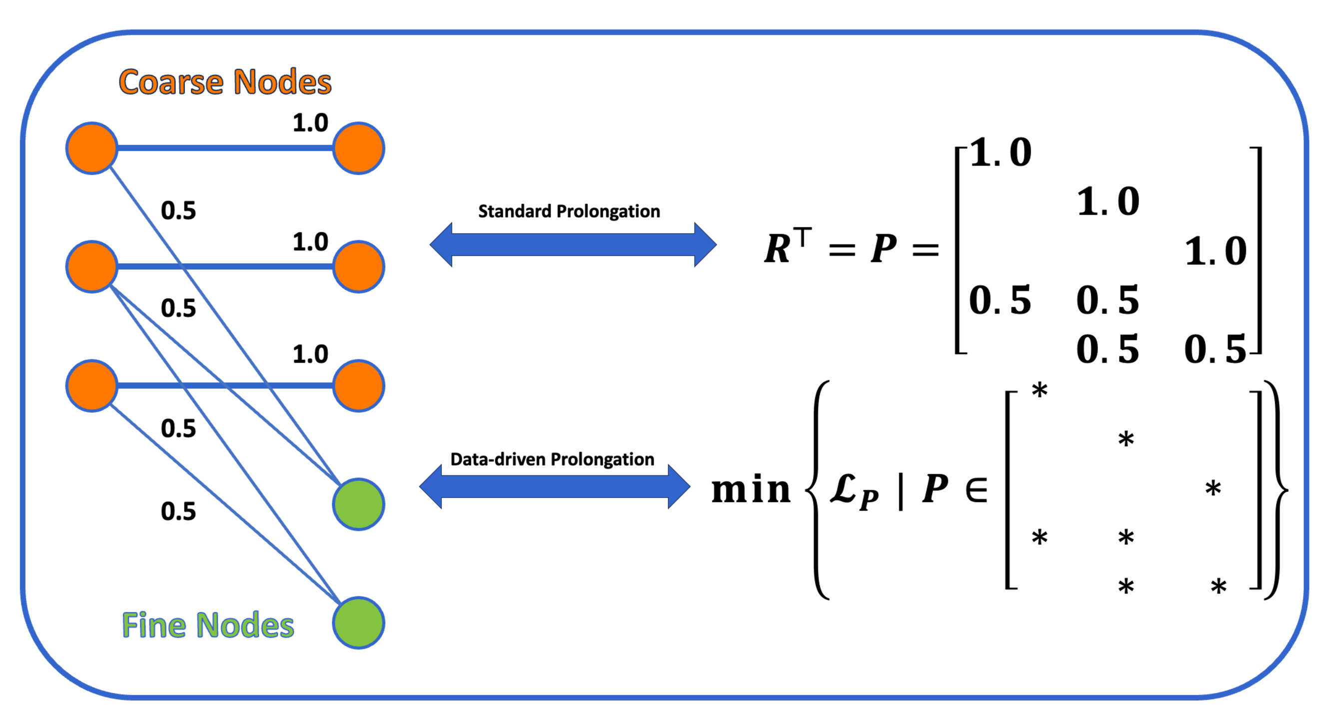

Figure 3: *Standard prolongation uses the weights from the matrix; however, using the same sparsity pattern, a neural network could find values that minimize a loss.*

RESULTS

Experiments: For a fixed smoother (Jacobi), compare standard MFEM and learned prolongation operator for 2D diffusion process with homogenous Dirichlet b.c.

$$\begin{cases} u_t - u_{xx} = 0, & u \in \Omega = [0,1]\times[0,1] \\ u = 0, & u \in \partial\Omega. \end{cases}$$

Set-up:

- 1000 epochs of ADAM with learning rate $\eta = 0.01$
- Graph attention with 4 layers and 4 heads
- 20 iterations of GMRES with tolerance $\tau = 10^{-10}$
- 5×5 grid with parallel refinement $\Longrightarrow$ 225×225
- 225 fine nodes and 49 coarse nodes

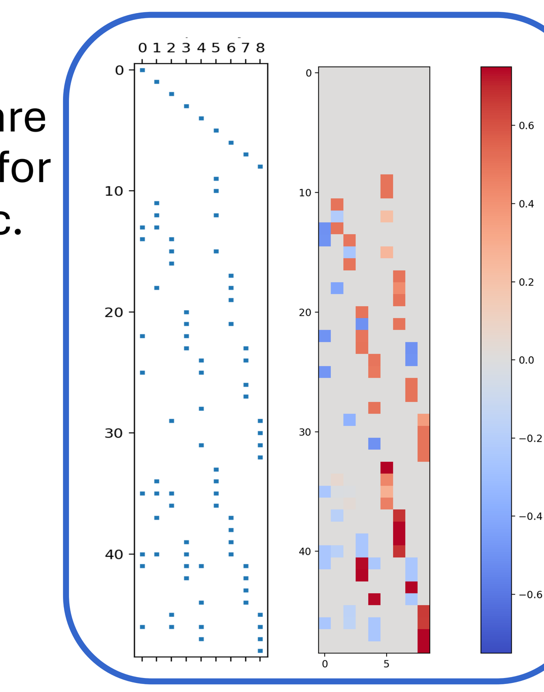

***Figure 4:* (Left)** *Fixed sparsity pattern for prolongation.* **(Right)** *Difference,* $P - P_{\text{True}}$.

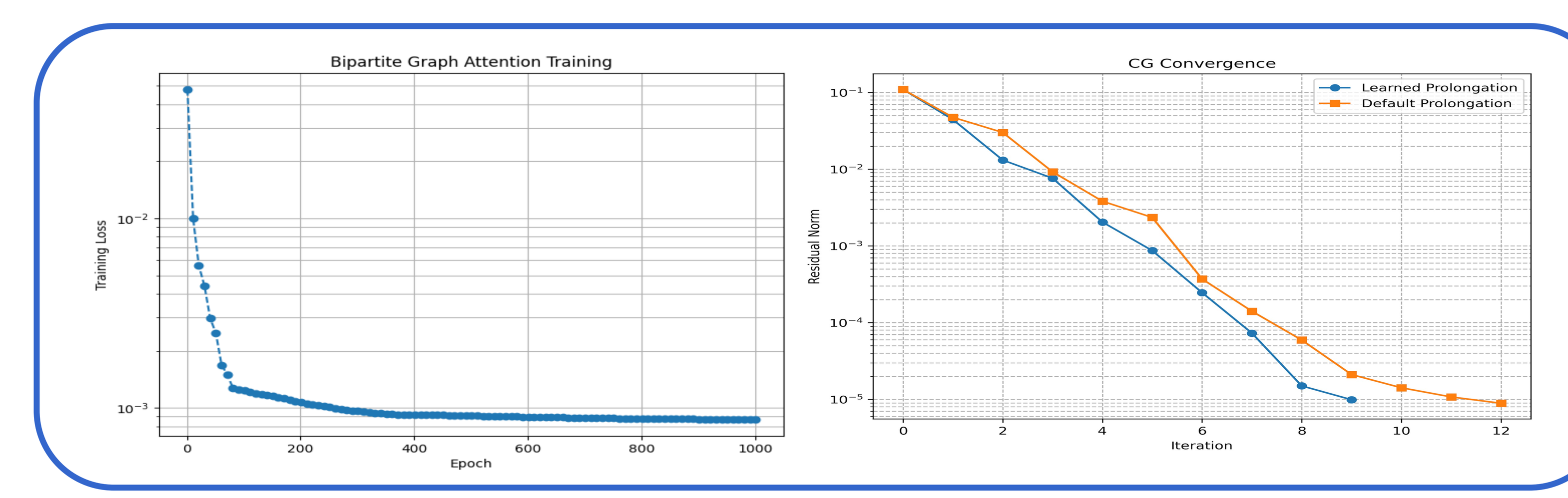

***Figure 5:* (Left)** *Training loss for bipartite GAT using,* $\mathcal{L}_P$. **(Right)** *Preconditioned GMRES convergence for* P, P_{True}.

FUTURE WORK

- Have GAT learn values and sparsity, so MFEM doesn't have to be run *a priori*
- Combine this with optimal data-driven smoother – either symmetric [2] or bipartite
- Extend results to different PDEs, such as anomalous diffusion, on unstructured and 3D meshes

REFERENCES

[1] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph Attention Networks*, in International Conference on Learning Representations, 2018.

[2] R. Huang, R. Li, and Y. Xi, *Learning Optimal Multigrid Smoothers via Neural Networks*, SIAM Journal of Scientific Computing, 45 (2022), pp 199-225

ACKNOWLEDGEMENTS

This work was partially supported by Lawrence Livermore National Laboratory under grants CASC LDRD 50962/24ERD033* and 38591/DSSI.Students. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Lawrence Livermore National Lab, the National Nuclear Security Administration, or the Department of Energy.

Lawrence Livermore National Laboratory
LLNL-POST-2008621
Prepared by LLNL under Contract DE-AC52-07NA27344.